AD NUMBER

AD059170

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

**Distribution authorized to U.S. Gov't. agencies and their contractors;
Administrative/Operational Use; 30 MAR 1955.
Other requests shall be referred to Office of Naval Research, Washington, DC 20360.**

AUTHORITY

onr ltr 9 nov 1977

FC

RELAXATION METHODS

by

George E. Forsythe

A section of a forthcoming book on

elasticity by I. S. Sokolnikoff, The McGraw-Hill Book Company

A Technical Report

Prepared under the sponsorship of the
Office of Naval Research
Project NR-044-144 and Contract Nonr-233(24)
and the
Office of Ordnance Research
Project TB2-0001(1210) and Contract DA 04-495-ORD-559
"Research in Numerical Analysis"

Numerical Analysis Research
Department of Mathematics
University of California
Los Angeles 24, California

March 30, 1955

125. Relaxation methods. We saw that the application of the method of finite differences may lead to the solution of certain systems of linear algebraic equations. Such systems frequently arise in applications, and a technique for solving them, bearing the name of the relaxation method, is the subject of this section.

Let us begin by considering a very simple example, the system (125.1), which corresponds to the problem of static equilibrium of ten equal masses, equally spaced on a light string under a uniform tension. (See Fig. 71, in which the sag is exaggerated.) By symmetry we assume $u_6 = u_5$, $u_7 = u_4$, etc. This example, together with other material in this section, is adapted from G. E. Forsythe's chapter in E. F. Beckenbach (editor), Modern Mathematics for the Engineer, The McGraw-Hill Book Company, 1955.*

---

*In recasting this material G. E. Forsythe wishes to acknowledge the sponsorship of the Office of Naval Research, project number NR-044-144 and contract Nonr-233(24); and the Office of Ordnance Research, project TB2-0001(1210) and contract DA 04-495-ORD-559.

$$(125.1) \qquad \begin{cases} -1 + 2u_1 - u_2 = r_1 = 0 \\ -1 - u_1 + 2u_2 - u_3 = r_2 = 0 \\ -1 - u_2 + 2u_3 - u_4 = r_3 = 0 \\ -1 - u_3 + 2u_4 - u_5 = r_4 = 0 \\ -1 - u_4 + u_5 = r_5 = 0 \end{cases}$$

The desired equilibrium position corresponds to the $u_1, \cdots, u_5$ obtained by solving (125.1) with $r_1 = \cdots = r_5 = 0$. A direct approach to solving the problem would be to solve (125.1) by the systematic elimination of unknowns, in the fashion of high-school algebra. The elimination might alternatively be expressed in terms of determinants. While these direct methods would solve (125.1) fairly readily, they would be very complicated indeed for a larger system like that corresponding to Figure 72.

An alternate solution of (125.1) has proved very popular among engineers and computers. We make a first guess of the u's, say $u_1 = 4$, $u_2 = 8$, $u_3 = 12$, $u_4 = 14$, $u_5 = 15$. From (125.1) we find that $r_1 = r_2 = -1$, $r_3 = 1$, $r_4 = r_5 = 0$. Since these residuals r_i are not all zero, we will improve the trial solution by reducing them. We pick one of the numerically largest r_i, say r_3, and bring it to zero by an appropriate change Δu_3 in u_3 alone. It is clear from (125.1) that a unit increase in u_3 ($\Delta u_3 = +1$) would cause changes only in r_2, r_3, and r_4, and these changes would be $\Delta r_2 = -1$, $\Delta r_3 = +2$, $\Delta r_4 = -1$. To make $r_3 = 0$ calls for $\Delta r_3 = -1$, which we bring about by selecting $\Delta u_3 = -.5$. As by-products we have $\Delta r_2 = \Delta r_4 = +.5$. Accumulating the r's and Δr's, we find the residuals $r_1 = -1.0$, $r_2 = -0.5$, $r_3 = 0.0$, $r_4 = 0.5$, $r_5 = 0.0$. There is now a single numerically largest residual, r_1, and we proceed to "liquidate" it by selecting $\Delta u_1 = +.5$. Next time $\Delta u_2 = +.5$, etc. Eight steps of this process are summarized in Table 1. An experienced computer goes very rapidly, calculating mentally and recording a residual only when it changes.

Table 1. Relaxation Solution of Equations (125.1)

	i •••	1	2	3	4	5	
First guess of u_i •••		4.0	8.0	12.0	14.0	15.0	
Residuals r_i •••••••		-1.0	-1.0	1.0	0.0	0.0	
i	Δu_i	Residuals					
3	-.5						
		-1.0	-0.5	0.0	0.5	0.0	
1	+.5						
		0.0	-1.0	0.0	0.5	0.0	
2	+.5						
		-0.5	0.0	-0.5	0.5	0.0	
4	-.3						
		- .5	.0	- .2	- .1	.3	
1	+.3						
		.1	- .3	- .2	- .1	.3	
2	+.2						
		- .1	.1	- .4	- .1	.3	
3	+.2						
		- .1	- .1	.0	- .3	.3	
5	-.3						
		- .1	- .1	.0	.0	.0	
Current solution ...		4.8	8.7	11.7	13.7	14.7	
Residuals		- .1	- .1	.0	.0	.0	Check
True solution		5.0	9.0	12.0	14.0	15.0	

Southwell[1] thinks of the r's as negatives of constraining forces actually applied to the weights to keep the system in equilibrium with the current displacements. Each step of the above calculation is then thought of as a relaxation of one of these external constraints. Hence Southwell's name for the process—relaxation.

At the bottom of Table 1 are cumulated the current values of the u's. E.g., $u_1 = 4 + .5 + .3 = 4.8$. Re-calculation of the residuals then confirms the computation so far. In these eight mental steps the maximum error of the u's has been reduced from 1.0 to 0.3. Further computing would improve the u's at a comparable rate, and it would not take long to achieve ordinary engineering accuracy.

There are many tricks used by relaxers. One, illustrated in Table 1, is to work to one significant digit only, and not to complicate the numbers by introducing overprecise corrections like $\Delta u_4 = -.25$. Thus residuals are liquidated only in the most significant digit. More precision comes automatically in later steps. Other tricks can be used to accelerate the convergence of the u's to the correct answers. Such acceleration is nearly always essential to solving a problem of any magnitude.

A great time-saver in engineering practice is not to draw up anything like Table 1, but instead to use a working drawing of the model as a computing sheet. The values of Δu_i and r_i can be recorded on the drawing.

Relaxation is really fun for a computer, for several reasons: (1) seeing the partial answer evolve lends a purpose to each step, and combats the usual tedium of day-long computing; (2) one's intelligence

is continually challenged by the possibility of improving the speed of convergence; (3) one need never waste much time in erroneous computing, as is possible in elimination.

There are many variations of relaxation methods. They all deal with solving systems of equations, usually linear, and they share these essential properties: (i) for any trial solution there is a measure of the error in each of the equations; (ii) for each unsatisfied equation there is a separate formula for improving the trial solution; (iii) one calculates at each step with the equation whose error is largest.

The relaxation method was originally devised for pencil-and-paper computing, without a keyboard calculator, and is ideally adapted to such work. It is reasonably adaptable to keyboard calculators, but here it seems to lose some of its relative superiority over other methods. For automatic digital computers, see below.

The relaxation method seems to date from Gauss,[2] who used and recommended the basic method and many of the standard tricks. Seidel[3] proved it would converge for linear systems with positive definite matrices. In the thirties Southwell[4] rediscovered Gauss's method, and named it. He and his school have developed the method and brought its wide applicability to the attention of engineers and scientists everywhere. The method has proved especially suited to the analysis of complicated redundant, pin-jointed frameworks. These have equations like (125.1), with more involved coefficients.

As a digression it should be noted that one-dimensional problems like that of Figure 71 should not in practice be solved by relaxation. For they are easily solved by a fast trial-and-recursion scheme, even for variable masses. For example, in (1) change the last equation to read

$$-1 \qquad -u_4 + 2u_5 - u_6 = r_5 = 0 \quad .$$

Fix $r_1 = \cdots = r_5 = 0$. By symmetry we should have $u_6 - u_5 = 0$. Try $u_1^{(1)} = 3$. From the modified equations (1), we find successively $u_2^{(1)} = 5$, $u_3^{(1)} = 6$, $u_4^{(1)} = 6$, $u_5^{(1)} = 5$, $u_6^{(1)} = 3$, whence $u_6^{(1)} - u_5^{(1)} = -2$. Now try $u_1^{(2)}$. We find successively $u_2^{(2)} = 11$, $u_3^{(2)} = 15$, $u_4^{(2)} = 18$, $u_5^{(2)} = 20$, $u_6^{(2)} = 21$, $u_6^{(2)} - u_5^{(2)} = +1$. Interpolating linearly between $u_1^{(1)}$ and $u_1^{(2)}$ to make $u_6 - u_5$ zero, we get $u_1 = 5$, whence the true solution is obtained recursively.

The point of Table 1 was to show the technique of relaxation in a simple setting. The practical applications of the method begin with two-dimensional problems—like trusses. Or, in closer relation to Figure 71, suppose one had an L-shaped network of light strings with 21 weights on it in a horizontal plane (Figure 72). How might one calculate the equilibrium position of the weights under large tension and under gravity? With the same assumptions as above, we find 21 equations like

$$(125.2) \qquad -1 - u_2 - u_6 + 4u_7 - u_8 - u_{12} = r_7 = 0 \quad .$$

It would be a tedious computation at a desk to solve these by elimination, and no simple recursive scheme works here. A practical answer is

relaxation, which works numerically about like Table 1, although with slower convergence. The same tricks apply as before.

An essential feature of systems of equations like (125.2) is that most of the coefficients of the u's are zero. It is this feature of such systems which makes relaxation a possible pencil-and-paper method of solving them.

The matrix A of coefficients a_{ij} of the system (125.1) or of the system of all 21 equations like (125.2) has two properties which will prove very important in our further discussion. First,

(125.3) A is symmetric and positive definite.

The second property, (125.4), concerns the geometry of the connecting strings in Figures 71 and 72. Note that the weights in the figures have been drawn in two colors: black and white. Note that each string connects weights of opposite color. Hence, in the equations the subscripts i of the unknowns u_i can be divided into two groups B, W (by color), so that

(125.4) $a_{ij} = 0$, for i in B, j in B ($i \neq j$) and for i in W, j in W ($i \neq j$).

Another way of expressing (125.4) comes from reordering the unknowns u_i and the corresponding equations so that the "blacks" entirely precede the "whites". Then the matrix takes the schematic form of Figure 73, where the circles denote zeros, the small crosses denote nonzero numbers, and the large crosses denote submatrices of zero and nonzero elements. Any system of linear equations satisfying (125.4) is said by Young[5] to have Property (A).

We note in passing that the first boundary-value problem for any second-order self-adjoint partial differential equation lacking a term in $\partial^2 u/\partial x \partial y$ leads to a symmetric linear system with Property (A), when difference equations are suitably introduced. If the partial differential equation is elliptic, then (125.3) holds.

Before we discuss methods suitable for electronic computers it will be convenient to introduce another method for solving a linear system. The most general system of n linear algebraic equations in n unknowns can be written in the form

$$(125.5) \qquad \sum_{j=1}^{n} a_{ij} u_j + b_i = r_i = 0 \qquad (i = 1, \cdots, n) \ .$$

For the moment we do not assume that the matrix satisfies (125.3) or (125.4), but it is essential that no $a_{ii} = 0$. Iterative methods for solving (125.5) have been popular since Gauss's time, if not longer. One process, called the Seidel or Gauss-Seidel method, is the following. One solves the first equation (125.5) for u_1, using the current values of $u_2, \cdots, u_n$. Then the second equation is solved for u_2, using the latest known values of $u_1, u_3, u_4, \cdots, u_n$. And so on. All the equations (125.5) are solved in cyclic order for $u_1, \cdots, u_n$, always with the latest values of the other unknowns. In other words, suppose $u_1^{(k)}, \cdots, u_n^{(k)}$ are known. One gets $u_1^{(k+1)}, \cdots u_n^{(k+1)}$ by successively solving these n equations:

$$(125.6) \qquad \sum_{j=1}^{i} a_{ij} u_j^{(k+1)} + \sum_{j=i+1}^{n} a_{ij} u_j^{(k)} + b_i = 0 \qquad (i = 1, 2, \cdots, n) .$$

One hopes that the $u_i^{(k)}$ will converge as $k \to \infty$ to the u_i which solve the system (125.5).

The reader will note that the Seidel process is closely related to the relaxation process described earlier in this section. The difference is the order in which the equations (125.5) are solved. In the relaxation method we do the equations in an order determined by the size of the r_i. In the Seidel process the order is fixed and cyclic.

Let us analyze the behavior of the Seidel process. In our applications the following theorem, apparently first completely proved by Schmeidler[6] in 1949, is essential:

If the matrix A is symmetric and positive definite, then in the Seidel process the $u_i^{(k)}$ converge as $k \to \infty$ to limits u_i $(i = 1, \cdots, n)$ solving the system (125.5).

What happens when A is not positive definite? Or, in any case, how does the vector $u^{(k)}$ approach u? If the convergence is slow, how can it be speeded up? The answers to these three questions can be obtained through some use of matrix theory, as follows.

Let the matrix of coefficients A in (125.5), assumed non-singular, be written as the sum of three matrices: $A = D + E + F$. Here D has the main diagonal of A, but is zero elsewhere; E has the below-diagonal elements of A, but is zero elsewhere; and F has the above-diagonal elements of A, but is zero elsewhere. Thus, schematically,

$$A = \begin{bmatrix} & F & \\ D & & \\ E & & \end{bmatrix} .$$

Equations (125.6) can be written in the following matrix-vector form:

$$(125.7) \qquad (D + E)u^{(k+1)} + F\,u^{(k)} + b = 0 \quad .$$

If u denotes the unique vector solving (125.5), we have

$$(125.8) \qquad (D + E)\,u + F\,u + b = 0 \quad .$$

Subtracting (125.8) from (125.7), and letting $e^{(k)} = u^{(k)} - u$ denote the error of $u^{(k)}$, we have

$$(125.9) \qquad (D + E)\,e^{(k+1)} + F\,e^{(k)} = 0 \quad .$$

Now since no $a_{ii} = 0$, the matrix $D + E$ has an inverse $(D + E)^{-1}$. Letting H denote the matrix $-(D + E)^{-1}F$, we find from (125.9) that

$$(125.10) \qquad e^{(k+1)} = H\,e^{(k)} \quad ,$$

whence

$$(125.11) \qquad e^{(k)} = H^{k}\,e^{(0)} \quad .$$

Equation (125.10) shows the linear behavior of the Seidel iteration process. A representation of the error $e^{(k)}$ in terms of the initial error is given by (125.11), on which a complete error analysis can be based. The Gauss-Southwell relaxation process is theoretically

more complicated just because it has no simple analog to (125.11).

From the theory of linear transformations we know when and how $e^{(k)}$ goes to zero. The two determinantal equations

$$(125.12) \qquad |H - \mu I| = 0 \quad \text{and} \quad |(D + E)\mu + F| = 0$$

have the same n real or complex roots $\mu_1, \cdots, \mu_n$. If all $|\mu_i| < 1$, then in the Seidel process $u^{(k)} \to u$. If any $|\mu_i| \geq 1$, the Seidel process diverges. In principle this settles the question of convergence.

For most matrices A, to each of the roots μ_i of (125.12) there corresponds a unique vector $y^{(i)}$ such that $Hy^{(i)} = \mu_i y^{(i)}$. That is, the transformation H leaves the vector $y^{(i)}$ unchanged in direction, but stretches it ($|\mu_i| > 1$) or shrinks it ($|\mu_i| < 1$) to the fraction μ_i of its previous length. All these vectors $y^{(i)}$ form an oblique coordinate system, in terms of which we can resolve the initial error vector $e^{(0)}$:

$$e^{(0)} = \sum_{i=1}^{n} c_i y^{(i)} .$$

Assume $|\mu_1| > |\mu_2| \geq \cdots \geq |\mu_n|$. After repeated multiplications by H, the resulting vector $H^k e^{(0)}$ is approximately moved into the direction corresponding to the root μ_1 of largest absolute value. Hence we find that

$$(125.13) \qquad e^{(k)} = H^k e^{(0)} \doteq c_1 \mu_1^k y^{(1)} ,$$

and we know how fast $e^{(k)} \to 0$. If $|\mu_1| < 1$, ultimately each step reduces the length of $e^{(k)}$ to the fraction $|\mu_1|$ of itself.

If $|\mu_1| < 1$, $e^{(k)} \to 0$ along one direction, that of $y^{(1)}$. Hence $u^{(k)} \to u$ along the direction of $y^{(1)}$. Cases where more than one $|\mu_1|$ dominate are more complicated, but can be treated with similar tools.

Knowing the geometric character of the convergence, it is not difficult to design acceleration processes to speed up the convergence of $u^{(k)}$ to u.

As an example of the Seidel process and its convergence, we use it to solve the system (125.1) with the same start as in Table 1. We have the following iterations:

$$\begin{cases} u_1^{(k+1)} = \frac{1}{2}\left(1 + u_2^{(k)}\right) \ ; \\ u_2^{(k+1)} = \frac{1}{2}\left(1 + u_1^{(k+1)} + u_3^{(k)}\right) \ ; \\ u_3^{(k+1)} = \frac{1}{2}\left(1 + u_2^{(k+1)} + u_4^{(k)}\right) \ ; \\ u_4^{(k+1)} = \frac{1}{2}\left(1 + u_3^{(k+1)} + u_5^{(k)}\right) \ ; \\ u_5^{(k+1)} = 1 + u_4^{(k+1)} \ . \end{cases}$$

Several rounds of this are shown in Table 2. The residuals are not shown, and one line of the table amounts to a full cycle of the above algorithm.

Table 2. Seidel Solution of Equations (125.1)

Ratio of worst errors	u	...	5	9	12	14	15
	$u^{(0)}$	...	4	8	12	14	15
.50							
	$u^{(1)}$	...	4.50	8.75	11.88	13.94	14.94
.24							
	$u^{(2)}$	...	4.88	8.88	11.88	13.91	14.91
.75							
	$u^{(3)}$	...	4.94	8.91	11.91	13.91	14.91
.96							
	$u^{(4)}$	...	4.955	8.932	11.921	13.916	14.916
.92							
	$u^{(5)}$	...	4.966	8.944	11.930	13.923	14.923
.909							
	$u^{(6)}$	...	4.972	8.951	11.937	13.930	14.930
.906							
	$u^{(7)}$	...	4.9755	8.9562	11.9431	13.9366	14.9366

In the first column of Table 2 is given the ratio of the worst error of the u's in the preceding and following rows of the table. By (125.13) this ratio converges to μ_1. (A solution of either equation (125.12) gives $\mu_1 = .9045$, $\mu_2 = .3455$, $\mu_3 = \mu_4 = 0$.) The approach of $u^{(k)}$ to u is one-sided and very regular. It will take about 22 cycles to gain one decimal point in accuracy. Making educated guesses at u in such problems is easy in desk work, if one knows (125.13) and its analog when $|\mu_1| = |\mu_2|$.

The present availability of electronic digital computing machines makes it possible to solve much larger problems than have been previously feasible. Such machines carry out arithmetic operations at an effective speed on the order of 10^4 times faster than a human being with a desk calculator. Something like 10^3 numbers of desk calculator precision can be held in a fast-access "memory" and made available as rapidly as the arithmetic organ can operate. Something like 10^4 more numbers can be held in an intermediate storage and transferred to the high-speed memory in a few milliseconds. Moreover, current developments will probably have made the figures given here obsolete before this book is published.

Because of the speed and capacity of such computers, many persons want to solve their problems on them. It is pertinent to ask, What methods will prove most feasible for the computers? While definitive answers must await investigations as yet undone, certain indications are now possible.

A first observation is that for large problems of the types of (125.1) or (125.2), iterative methods are relatively attractive, for much the same reasons as for pencil-and-paper calculation. But the relaxation method as outlined in connection with (125.1) has one considerable disadvantage. The scanning of all the residuals r_i in a search for $\max_i |r_i|$ is comparatively time-consuming. In fact, while computing r_i it would take almost no extra time to solve the i-th equation for u_i. But if one solves the i-th equation for u_i ($i = 1, \cdots, n$), one is actually carrying out the Seidel process, which is accordingly preferred in machine calculation to conventional relaxation.

A second observation is that solving a large system (125.5) by the Seidel method is likely to be slow. To speed up the solution, acceleration methods are needed, as indicated above. But accelerations involve new routines, new coding, and the mundane but important problems of storing or reading in new codes. It is important with machines to reduce coding and operating to the utmost in simplicity.

It is the remarkable discovery of Young[7] that for certain problems a modification of the Seidel process will vastly speed up the convergence, while scarcely complicating the coding at all. Where applicable, it thus eliminates the necessity for special acceleration routines.

It has long been observed by relaxers that the Gauss-Southwell process usually goes faster if one "overrelaxes" a little at each step. Young was therefore convinced of the value of analyzing overrelaxation carefully in conjunction with the Seidel process. He confines himself to positive definite symmetric matrices with Property (A); see (125.4). In each step of the Seidel process (125.6) or (125.7), Young suggests that one first compute the Seidel value — call it $v_i^{(k)}$ — and then compute

$$u_i^{(k+1)} = u_i^{(k)} + \beta(v_i^{(k)} - u_i^{(k)}) \quad .$$

This amounts to an overrelaxation of $100(\beta - 1)$ per cent. Young asks which choice of β $(1 < \beta < 2)$ is best.

The analysis proceeds much as in the Seidel process, since this systematic overrelaxation process is also a linear one. We have

$$(125.14) \qquad Eu^{(k+1)} + Dv^{(k)} + Fu^{(k)} + b = 0$$

and

$$u^{(k+1)} = u^{(k)} + \beta[v^{(k)} - u^{(k)}] = (1 - \beta)\, u^{(k)} + \beta v^{(k)} \quad .$$

We now eliminate $v^{(k)}$ from (125.14). Since

$$Du^{(k+1)} = (1 - \beta)\, Du^{(k)} + \beta Dv^{(k)} \quad ,$$

we have

$$(125.15) \qquad Eu^{(k+1)} + \frac{1}{\beta} Du^{(k+1)} + \left(1 - \frac{1}{\beta}\right) Du^{(k)} + Fu^{(k)} + b = 0 \quad .$$

Equation (125.15) describes systematic overrelaxation, just as equation (125.7) describes the Seidel process. (Note that (125.15) reduces to (125.7) for $\beta = 1$.) The speed of the convergence of Young's process is measured by the largest in modulus of the roots σ_i of the following determinantal equation, analogous to the second part of (125.12):

$$(125.16) \qquad \left|\sigma E + \frac{\sigma}{\beta} D + \left(1 - \frac{1}{\beta}\right) D + F\right| = 0 \quad .$$

For matrices satisfying (125.3) and (125.4) and for a certain ordering of the equations, it can be shown that the maximum of the $|\sigma_i|$ is least when we choose $\beta = 2(1 + \sqrt{1 - \mu_1})^{-1}$, where μ_1 is the largest root of (125.12). Hence this β defines the optimal amount to overrelax. Moreover, for this β all $|\sigma_i|$ are equal.

To illustrate the method, we show in Table 3 the result of solving (125.1) by systematic overrelaxation. Corresponding to $\mu_1 = .9045$, we take $\beta = 1.528$.

Table 3. Solution of (125.1) by Systematic Overrelaxation

Ratio of worst errors	u	5	9	12	14	15
	$u^{(0)}$	4	8	12	14	15
.34						
	$u^{(1)}$	4.76	9.34	12.26	14.20	15.31
.51						
	$u^{(2)}$	5.39	9.32	12.26	14.33	15.34
.54						
	$u^{(3)}$	5.039	9.060	12.161	14.209	15.140
.76						
	$u^{(4)}$	5.025	9.110	12.159	14.119	15.108
.74						
	$u^{(5)}$	5.071	9.118	12.097	14.093	15.085
.52						
	$u^{(6)}$	5.053	9.052	12.059	14.061	15.048
.59						
	$u^{(7)}$	5.012	9.027	12.036	14.032	15.024
.67						
	$u^{(8)}$	5.014	9.024	12.024	14.020	15.018
.58						
	$u^{(9)}$	5.011	9.014	12.013	14.013	15.010
.56						
	$u^{(10)}$	5.0049	9.0063	12.0079	14.0068	15.0051

The same ratio of worst errors is given as in Table 2. It is irregular in Table 3, but will converge to $\beta - 1 = .528$. With this value, one will require only 3.6 cycles per decimal point. Thus the value $\beta = 1.528$ is much superior to the value $\beta = 1$ of the Seidel method.

G. E. Forsythe has coded for SWAC[8] the Young method for a simple difference equation,

(125.17)

$$u(x + h, y) + u(x, y + h) + u(x - h, y) + u(x, y - h) - 4\,u(x, y) = 0 \quad ,$$

corresponding to the Laplace differential equation. The code will accommodate a network as large as 32 by 128 points. The shape of the boundary is immaterial, except that the boundary points must lie on the nodes of the network. The following example may illustrate the usefulness of the method.

For a rectangle of $30 \times 68 = 2040$ interior unknown points, each cycle of relaxation takes 8.5 seconds. For the Seidel process the dominant eigenvalue $\mu_1 = 0.99416$. To reduce the error to 10^{-6} times its initial value by the Seidel process ($\beta = 1$) would require about 2300 cycles, because approximately one has $(.99416)^{2300} = 10^{-6}$. This would take over five hours on SWAC. If, however, β is taken at its optimal value of $2(1 + (1 - .99416)^{\frac{1}{2}})^{-1} = 1.85802$, then the dominant eigenvalue is $\sigma_1 = .85802$. For this σ_1, it requires about 90 iterations, accomplished in only 13 minutes, to reduce the error from 1 to 10^{-6}.

Some practice with SWAC convinces people it is not difficult to estimate β well enough in a few minutes. Actual running time to reduce the error by a factor of 10^{-6} is on the order of 20 to 30 minutes, including the time necessary to determine β approximately. Similar experience is reported by Young and Lerch.[9]

On SWAC the progress of the calculation can be monitored by observing the value of $E^{(k)} = \sum_{i=1}^{n} |r_i^{(k)}|$ as the calculation proceeds. When $\beta = 1$, $E^{(k)}$ decreases monotonically and smoothly. When β is around the optimal value, $E^{(k)}$ has comparatively wild fluctuations, probably because of "beats" (something of this is seen in Table 3) between the various complex frequencies σ_i of equal magnitude. It is perhaps also because the operator carrying $E^{(k)}$ into $E^{(k+1)}$ has a nonlinear elementary divisor.

For recent research on Young's and similar processes, see Riley[10] and Sheldon.[11] There is evidence[12] that systematic overrelaxation is also useful for some matrices not having Property (A). The Seidel method, Young's modification, and the SWAC codes are adaptable to obtaining the fundamental eigenvalue of a matrix with Property (A).

March 30, 1955

FOOTNOTES FOR SECTION 125

[1]Southwell, R. V.: "Relaxation Methods in Engineering Science," Oxford University Press, 1940, 252 pp.

[2]Gauss, C. F.: Brief an Gerling, 26 December 1823, "Werke," Vol. 9, pp. 278-281. [Translated by G. E. Forsythe, Math. Tables and Other Aids to Computation, Vol. 5 (1951), pp. 255-258.]

[3]Seidel, Ludwig: Ueber ein Verfahren, die Gleichungen, auf welche die Methode der kleinsten Quadrate führt, sowie lineäre Gleichungen überhaupt, durch successive Annäherung aufzulösen, Abh. Math. Phys. Kl., Bayrische Akad. Wiss., München, Vol. 11 (1874), pp. 81-108.

[4]Southwell, R. V.: op. cit. and "Relaxation Methods in Theoretical Physics," Clarendon Press, Oxford, 1946, 248 pp.

[5]Young, David: Iterative Methods for Solving Partial Difference Equations of Elliptic Type, Trans. Amer. Math. Soc., Vol. 76 (1954), pp. 92-111. (Condensation of his 1950 Harvard thesis.)

[6]Schmeidler, Werner: "Vorträge über Determinanten und Matrizen mit Anwendungen in Physik und Technik," Akademie-Verlag, Berlin, 1949, 155 pp. Professor A. Ostrowski has traced incomplete proofs back to P. Pizzetti, Atti della Reale Accademia dei Lincei. Rendiconti (4) Vol. 3_2(1887), pp. 230-235, 288-293.

[7]Op. cit., footnote 5. The same suggestion was put forward a little earlier in a special case by Frankel, Stanley P.: Convergence Rates of Iterative Treatments of Partial Differential Equations, Math. Tables and Other Aids to Computation, Vol. 4 (1950), pp. 65-75.

[8]National Bureau of Standards Western Automatic Computer, located in the U.C.L.A. Department of Mathematics.

[9]Young, David and Lerch, Francis: The Numerical Solution of Laplace's Equation on ORDVAC, Ballistics Research Laboratories Memorandum Report No. 708, Aberdeen Proving Ground, July 1953, 25 pp.

[10]Riley, J. D.: Iteration procedures for the Dirichlet difference problem, Math. Tables and Other Aids to Computation, Vol. 8 (1954), pp. 125-131.

[11]Sheldon, J. W.: On the numerical solution of elliptic difference equations, Math. Tables and Other Aids to Computation, Vol. 9 (1955), pp. 000-000.

[12]Charney, J. G. and Phillips, N. A.: Numerical integration of the quasi-geostrophic equations for barotropic and simple baroclinic flows, J. Meteorology, Vol. 10 (1953), pp. 71-99.

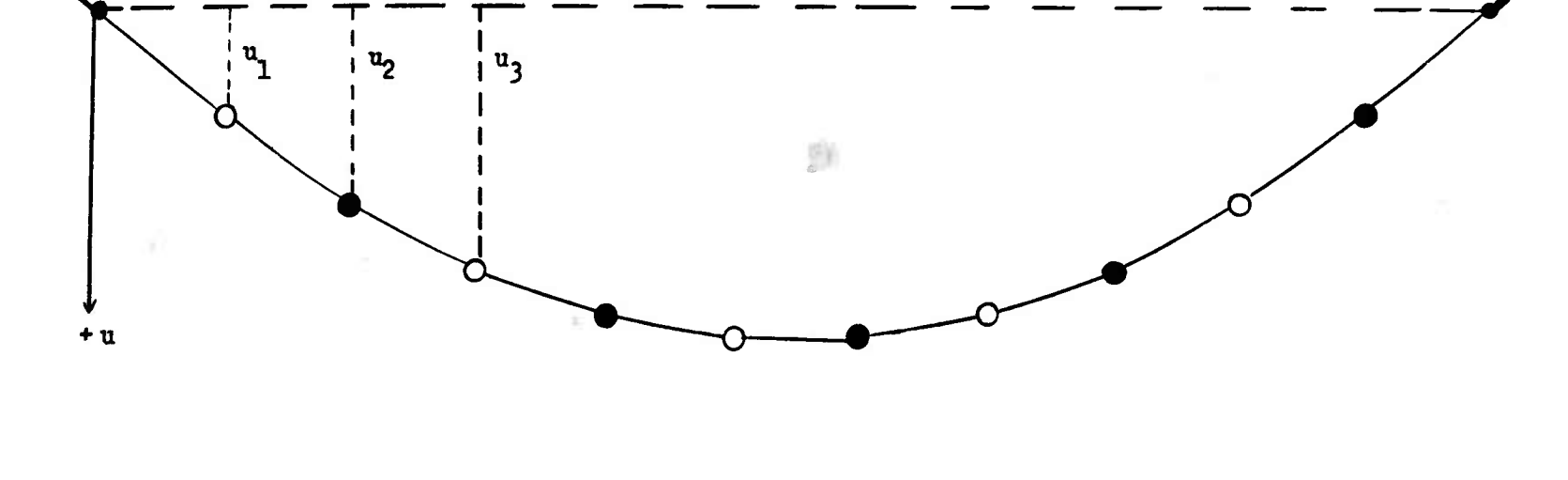

Figure 71

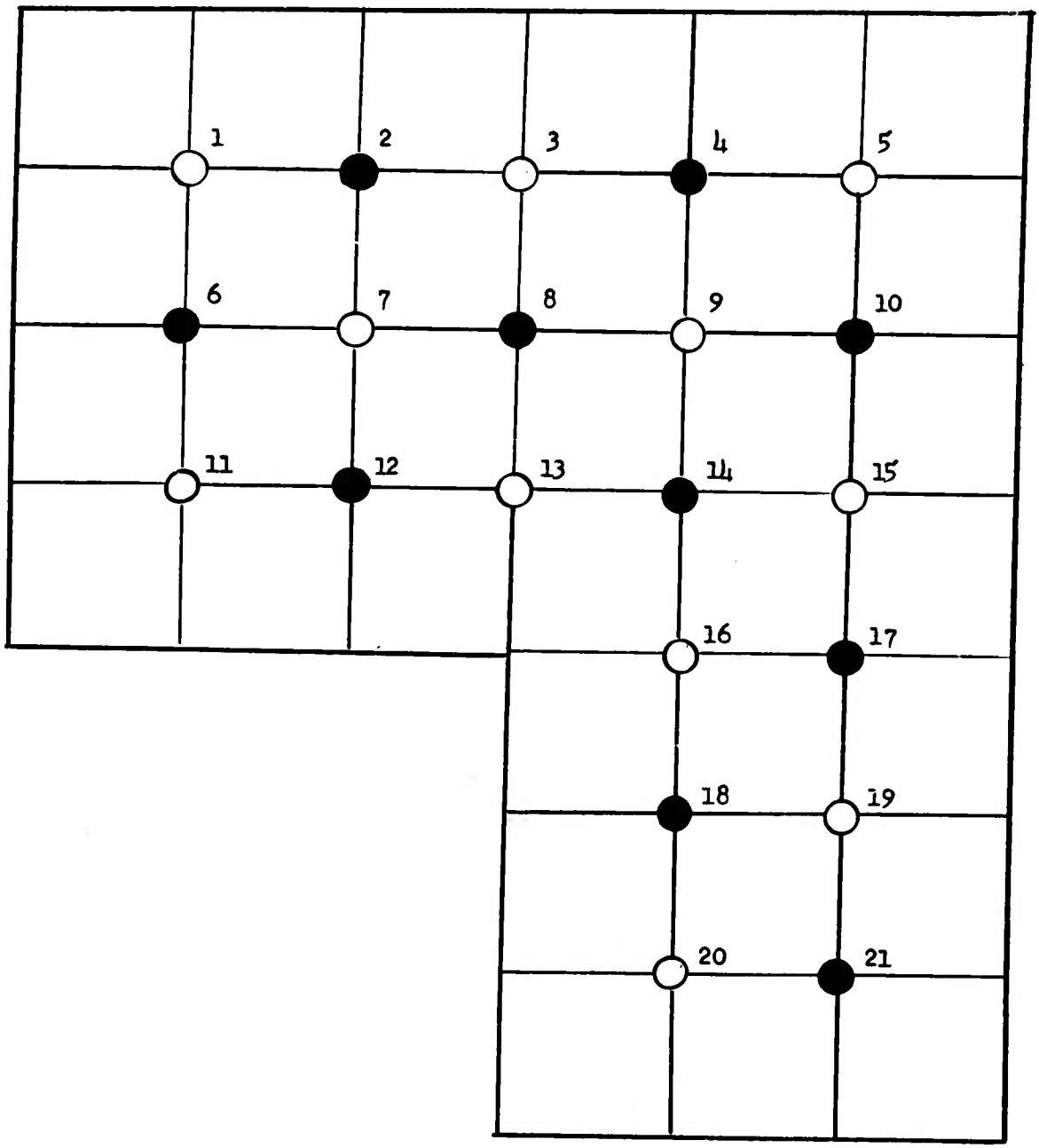

Figure 72

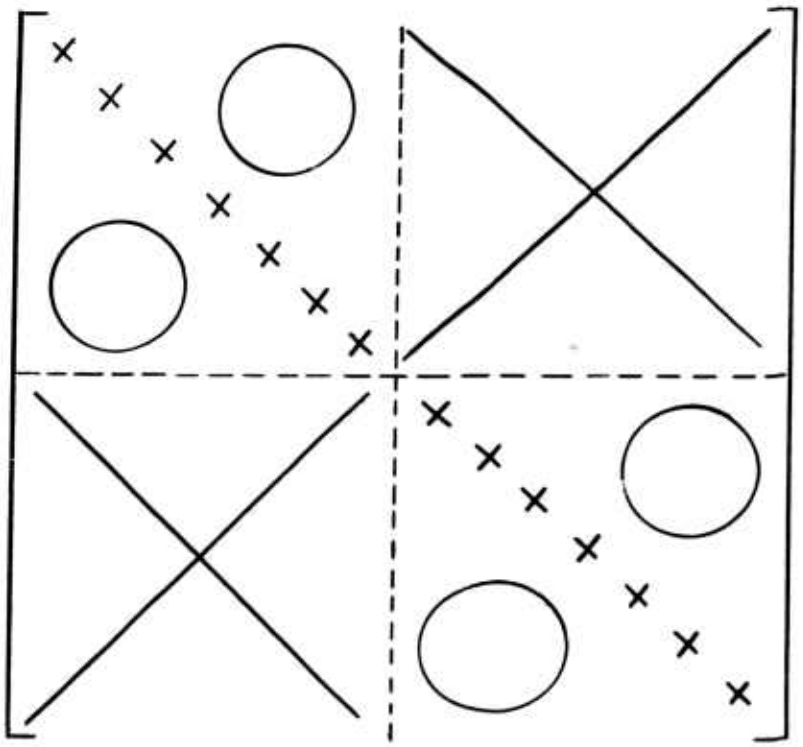

Figure 73